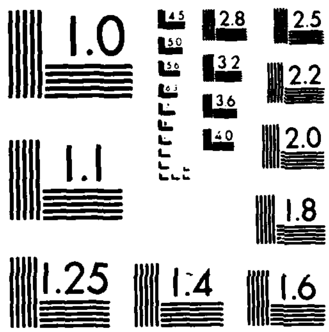
AD-A143 439 THE LEAST SQUARES LATTICE ALGORITHM: AN ALTERNATE DERIVATION WITH A DISCU..(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA S KALSON JAN 84 NOSC/TR-950 1/1

UNCLASSIFIED F/G 12/1 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

AD-A143 439

NOSC TR 950

Technical Report 950

THE LEAST SQUARES LATTICE ALGORITHM

An alternate derivation with a discussion of numerical conditioning

S. Kalson

January 1984
Final Report
November 1983 — January 1984

Prepared for
Naval Electronic Systems Command

NOSC

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152

NAVAL OCEAN SYSTEMS CENTER SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

J.M. PATTON, CAPT, USN
Commander

R.M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

The work reported herein was conducted for the Naval Electronic Systems Command, under program element OMN, subtask HIDRA, over the period November 1983 - January 1984.

Released by
E. R. Jahn, Head
Submarine Communications
Systems Division

Under authority of
H. F. Wong, Acting Head
Communications Systems and
Technology Department

AS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NOSC TR 950	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Ocean Systems Center; 6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) San Diego, CA 92152	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Electronic Systems Command; 8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code) Washington, DC 20363	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: OMN; PROJECT NO: HIDRA; TASK NO: ; WORK UNIT NO:

11. TITLE (include Security Classification)

THE LEAST SQUARES LATTICE ALGORITHM
An Alternate Derivation with a discussion of numerical conditioning

12. PERSONAL AUTHOR(S)
Seth Kalson

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM Nov 1983 TO Jan 1984	January 1984	47

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Least squares lattice; Lattice algorithm; Orthogonal triangularization; Cholesky decomposition; Gram-Schmidt orthogonalization; Condition number

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A new derivation of the least squares lattice algorithm is given in which the filter coefficients are solved for directly by a Gram-Schmidt orthogonalization of the data. This approach shows that the unnormalized and normalized least squares lattice algorithms have the same numerical conditioning as the so-called normal equations associated with least squares problems. Thus, contrary to some of the literature, the normalized lattice is not numerically superior to the unnormalized lattice for ill-conditioned problems.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION	Agency Accession
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (include Area Code)	22c. OFFICE SYMBOL

DD FORM 1473, 84 JAN

83 APR EDITION MAY BE USED UNTIL EXHAUSTED
ALL OTHER EDITIONS ARE OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

CONTENTS

1. INTRODUCTION

In the literature there appear to be two different approaches to the derivation of the least squares lattice algorithms: an algebraic and a geometric approach. The first method chronologically precedes the former and was originally presented by Morf, Lee, et al. (ref 1). See ref 2 for a clear and precise development. The geometrical approach (ref 3, 4) gives more insight and is much more powerful in the sense that it encompasses an entire class of computationally efficient algorithms, (ref 4).

The derivation presented in this report is different than the geometrical approach in that the filter coefficients are solved for directly by Gram-Schmidt orthogonalization of the data (ref 5, 6). The advantage of this approach is to provide insight into the numerical conditioning of the algorithm, which is shown to be equivalent to the numerical conditioning of solving the so-called normal equation associated with least squares. Although only the unnormalized least squares lattice is derived explicitly, the normalized version is briefly discussed and is easily shown to have the same conditioning as the unnormalized version. A simple numerical example is given to illustrate this point. Thus contrary to popular belief, the normalized least squares lattice is not numerically superior to the unnormalized algorithm for ill-conditioned problems.

2. DEFINITIONS

$x(t)$ and $y(t)$ shall represent two, complex, scalar discrete time series. The pre-window case will be assumed, i.e.,

$$x(t) = y(t) = 0 \qquad t < 0 \tag{1}$$

Let $\bar{y}(T)$ be a T by 1 column vector whose t^{th} component is given by

$$[\bar{y}(T)]_t = y(t) \qquad t = 0,1,\cdots T-1 \tag{2}$$

Also, let $\bar{x}^i(T)$ be a T by 1 column vector given by

$$[\bar{x}^i(T)]_t = \begin{cases} x(t-i) & i \leq t \leq T-1 \\ 0 & 0 \leq t < i \end{cases} \tag{3}$$

where i is an integer.

For any two vectors, $\bar{u}(T)$, $\bar{v}(T)$, of dimension T, we define the inner product

$$<\bar{u}(T),\bar{v}(T)> = \sum_{t=0}^{T-1} [\bar{u}(T)]_t^* [\bar{v}(T)]_t \tag{4}$$

This is the familiar inner product of complex vectors. We also have the Euclidian norm

$$||\bar{u}(T)||^2 = <\bar{u}(T), \bar{u}(T)> \tag{5}$$

The following T by T diagonal matrix is needed

$$\Lambda(T) = \begin{bmatrix} \lambda^{T-1} & & & 0 \\ & \lambda^{T-2} & & \\ & & \ddots & \\ & & & \lambda \\ & 0 & & & 1 \end{bmatrix} \tag{6}$$

where $0 \leq \lambda \leq 1$. Let $\Lambda^{\frac{1}{2}}(T)$ be such that

$$\Lambda^{\frac{1}{2}}(T)\Lambda^{\frac{1}{2}}(T) = \Lambda(T) \tag{7}$$

Then it is quite obvious that $\Lambda^{\frac{1}{2}}(T)$ is the diagonal matrix

$$\Lambda^{\frac{1}{2}}(T) = \begin{bmatrix} \lambda^{(T-1)/2} & & & & 0 \\ & \lambda^{(T-2)/2} & & & \\ & & \ddots & & \\ & & & \lambda^{\frac{1}{2}} & \\ 0 & & & & 1 \end{bmatrix} \tag{8}$$

We will need the following two properties of $\Lambda^{\frac{1}{2}}(T)$

$$\Lambda^{\frac{1}{2}}(T+1) = \left[\begin{array}{c|c} \lambda^{T/2} & 0 \\ \hline 0 & \Lambda^{1/2}(T) \end{array}\right] \tag{9a}$$

$$= \left[\begin{array}{c|c} \lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}(T) & 0 \\ \hline 0 & 1 \end{array}\right] \tag{9b}$$

Let the T by 1 column vectors $\bar{x}^i(T)$ and $\bar{y}(T)$ be

$$\bar{\mathbf{x}}^i(t) = \Lambda^{\frac{1}{2}}(T)\ \bar{x}^i(T) \tag{10a}$$

$$\bar{\mathbf{y}}(T) = \Lambda^{\frac{1}{2}}(T)\ \bar{y}(T) \tag{10b}$$

It should be clear from the definitions that

$$[\bar{\mathbf{x}}^i(T)]_T = x(T-1) \tag{11a}$$

$$[\bar{\mathbf{y}}^i(T)]_T = y(T-1) \tag{11b}$$

These properties will be useful later. We also define the T by N matrix $\theta^N(T)$ as

$$\theta^N(T) = [\bar{\mathbf{x}}^o(T)\ \bar{\mathbf{x}}^1(T)\cdots\bar{\mathbf{x}}^{N-1}(T)] \tag{12}$$

3. PROBLEM STATEMENT

We seek the N by 1 column vector $\bar{s}^N$ that minimizes the weighted sum of squares

$$\left\| \begin{bmatrix} y(0) \\ \cdot \\ \cdot \\ \cdot \\ y(T) \end{bmatrix} - \begin{bmatrix} x(0) & 0 & \cdots & 0 \\ \vdots & x(0) & \cdots & 0 \\ \vdots & \cdot & \cdots & x(0) \\ \vdots & & & \vdots \\ x(T) & x(T-1) & \cdots & x(T-N+1) \end{bmatrix} \bar{s}^N \right\|^2_{\Lambda(T+1)} \tag{13}$$

The weight matrix $\Lambda(T+1)$, for $\lambda < 1$, allows the filter to be adaptive by "forgetting" old data. Let $\bar{h}^N(T)$ be the $\bar{s}^N$ that minimizes the above expression and denote the minimum by $J_N(T)$. Thus, using the definitions of Section 2, we can write the least squares problem as

$$J_N(T) = \min_{\bar{s}^N} \|\bar{y}(T+1) - \theta^N(T+1)\bar{s}^N\|^2 = \|\bar{y}(T+1) - \theta^N(T+1)\bar{h}^N(T)\|^2 \tag{14}$$

For many applications, such as channel equalization with a known sequence, the filtered output $\sum_{i=0}^{N-1} x(T-i)[\bar{h}^N(T)]_i$ is of interest. However for other applications, such as noise cancellation or decision-directed equalization, the filtered output $\sum_{i=0}^{N-1} x(T-i)[\bar{h}^N(T-1)]_i$ is required (ref 8, 9). Therefore, we will define the two error residuals

$$\varepsilon_N(T) = y(T) - [\, x(T) \cdots x(T-N+1)]\bar{h}^N(T) \tag{15a}$$

$$\varepsilon'_N(T) = y(T) - [\, x(T) \cdots x(T-N+1)]\bar{h}^N(T-1) \tag{15b}$$

A computationally efficient algorithm shall be derived for the time and order updates of $\varepsilon_N(T)$ and $\varepsilon'_N(T)$. The algorithm is efficient in the sense that it calculates $\varepsilon_N(T)$ and $\varepsilon'_N(T)$ for all orders $N=0,1,2,\cdots N_o$ with only on the order of N_o calculations per time update. However, before equation (14)

can be solved, we must solve the auxilary problem of the weighted least squares one-step forward and backward predictor filter, the so-called least squares lattice algorithm.

4. DERIVATION OF THE LEAST SQUARES LATTICE ALGORITHM

First, the general solution to the forward and backward filtering problem, along with some useful properties, shall be presented in sections 4.A. and 4.B. Order updates are derived in 4.C. and 4.D., followed by the time updates in 4.E. The least squares lattice algorithms are summarized in 5.

4.A. THE ONE-STEP FORWARD PREDICTOR

The predictor filter is found by minimizing:

$$\left\| \begin{bmatrix} x(0) \\ \vdots \\ \vdots \\ x(T) \end{bmatrix} - \begin{bmatrix} 0 & & 0 \\ x(0) & & \vdots \\ \vdots & & x(0) \\ \vdots & & \vdots \\ x(T-1) & & x(T-N) \end{bmatrix} \bar{s}^{N} \right\|^{2}_{\Lambda(T+1)} \tag{16}$$

with respect to $\bar{s}^N$. By property (9a), the above norm is equivalent to

$$||\bar{x}^{-1}(T) - \theta^N(T)\bar{s}^N||^2 + \lambda^T|x(0)|^2 \tag{17}$$

Let the $\bar{s}^N$ that achieves this minimum be $\bar{f}^N(T)$, with minimum norm $J^f_N(T)$. Thus,

$$J^f_N(T) = ||\bar{x}^{-1}(T) - \theta^N(T)\bar{f}^N(T)||^2 + \lambda^T|x(0)|^2 \tag{18}$$

The minimization of (17) can easily be solved by finding a T by T unitary matrix Q(T), such that

$$Q^{\dagger}(T)\theta^N(T) = \begin{matrix} N \\ T-N \end{matrix}\left\{\begin{bmatrix} W^N(T) \\ \text{-----} \\ 0 \end{bmatrix}\right. \tag{19}$$

where $W^N(T)$ is a N by N, nonsingular, upper triangular matrix (see ref 5 and 6). The dagger, "†", represents the complex conjugate transpose. Q(T) exists, provided $\theta^N(T)$ is full rank for $N \leq T$. Thus, $\{\bar{x}^o(T) \cdots \bar{x}^{T-1}(T)\}$ must be linearly independent set of vectors.

To see how (19) solves (17), we make use of the unitary property of Q(T), i.e., $Q^\dagger(T)\, Q(T) = Q(T)\, Q^\dagger(T) = I$, to write (17) as

$$||Q^\dagger(T)\bar{x}^{-1}(T) - Q^\dagger(T)\theta^N(T)\bar{s}^N||^2 + \lambda^T|x(0)|^2 \tag{20}$$

Using (19), we can split the above norm into

$$||\bar{x}_N^{-1'}(T) - W^N(T)\bar{s}^N||^2 + ||\bar{x}_N^{-1''}(T)||^2 + \lambda^T|x(0)|^2 \tag{21}$$

where,

$$Q^\dagger(T)\bar{x}^{-1}(T) = \begin{array}{l} N\left\{\vphantom{\bar{x}_N^{-1'}}\right. \\ \\ T\text{-}N\left\{\vphantom{\bar{x}_N^{-1''}}\right. \end{array} \begin{bmatrix} \bar{x}_N^{-1'}(T) \\ \text{-------} \\ \bar{x}_N^{-1''}(T) \end{bmatrix} \tag{22}$$

Minimization of (21) is now simple, with $\bar{f}^N(T)$ and $J_N^f(T)$ given by

$$W^N(T)\bar{f}^N(T) = \bar{x}_N^{-1'}(T) \tag{23a}$$

$$J_N^f(T) = ||\bar{x}_N^{-1''}(T)||^2 + \lambda^T|x(0)|^2 \tag{23b}$$

Let $\{\bar{q}^o(T) \cdots \bar{q}^{T-1}(T)\}$ be found by a Gram-Schmidt orthogonalization of $\{\bar{x}^o(T) \cdots \bar{x}^{T-1}(T)\}$; then Q(T) is given by

$$Q(T) = [\bar{q}^o(T) \cdots\cdots \bar{q}^{T-1}(T)] \tag{24}$$

This can be seen as follows. The Gram-Schmidt procedure is given by

$$\bar{q}^{o}(T) = \frac{\bar{x}^{o}(T)}{||\bar{x}^{o}(T)||^{2}} \tag{25a}$$

For k = 1, T-1, do

$$\tilde{\bar{q}}^{k}(T) = \bar{x}^{k}(T) - \sum_{j=0}^{k-1} \langle \bar{q}^{j}(T), \bar{x}^{k}(T) \rangle \, \bar{q}^{j}(T) \tag{25b}$$

$$\bar{\bar{q}}^{k}(T) = \frac{\tilde{\bar{\bar{q}}}^{k}(T)}{||\tilde{\bar{q}}^{k}(T)||^{2}} \tag{25c}$$

End loop.

This procedure produces a set of orthonormal vectors $\{\bar{q}^{o}(T)\cdots\bar{q}^{N}(T)\}$ that span the same space as $\{\bar{x}^{o}(T)\cdots\bar{x}^{N}(T)\}$ for N = 0,$\cdots$T-1. Denote this space by

$$S^{N}(T) = \{\bar{q}^{o}(T)\cdots\bar{q}^{N}(T)\} = \{\bar{x}^{o}(T)\cdots\bar{x}^{N}(T)\} \tag{26}$$

Also, we have the additional property

$$\langle \bar{q}^{i}(T), \bar{x}^{j}(T) \rangle = 0 \qquad j < i \tag{27}$$

Thus, from (27) and (12), we have (19) where $W^{N}(T)$ is given by

$$[W^{N}(T)]_{ij} = \langle \bar{q}^{i}(T), \bar{x}^{j}(T) \rangle \tag{28}$$

$W^{N}(T)$ is upper triangular because of (27). Also, Q(T) is unitary by the orthonormal property of $\{\bar{q}^{o}(T)\cdots\bar{q}^{T-1}(T)\}$.

It will also be necessary to consider the vector

$$\bar{F}^N(T) = \theta^N(T)\bar{f}^n(T) \tag{29}$$

$\bar{F}^N(T)$ can be interpreted as the projection of $\bar{x}^{-1}(T)$ onto $S^{N-1}(T)$. This is easily seen as follows:

From (19),

$$Q^\dagger(T)\bar{F}^N(T) = Q^\dagger(T)\theta^N(T)\bar{f}^N(T) = \begin{bmatrix} W^N(T) \\ \text{-----} \\ 0 \end{bmatrix} \bar{f}^n(T)$$

$$= \begin{bmatrix} W^N(T)\bar{f}^N(T) \\ \text{-----------} \\ 0 \end{bmatrix} \tag{30}$$

From (23a), we write (30) as

$$Q^\dagger(T)\bar{F}^N(T) = \begin{bmatrix} \bar{x}^{-1'}_N(T) \\ \text{---------} \\ 0 \end{bmatrix} \tag{31}$$

Since Q(T) is unitary, multiply (31) on the left by Q(T) to obtain

$$\bar{F}^N(T) = Q(T) \begin{bmatrix} \bar{x}^{-1'}_N(T) \\ \text{-------} \\ 0 \end{bmatrix} \tag{32}$$

From the definition of Q(T) and $\bar{x}^{-1'}_N(T)$, equations (24) and (22), we have

$$\bar{F}^N(T) = [\bar{q}^o(T)\cdots\bar{q}^{T-1}(T)] \begin{bmatrix} <\bar{q}^o(T),\bar{x}^{-1}(T)> \\ \vdots \\ <\bar{q}^{N-1}(T),\bar{x}^{-1}(T)> \\ \text{-----------------} \\ 0 \end{bmatrix} \tag{33}$$

This is equivalent to

$$\bar{F}^N(T) = \sum_{i=0}^{N-1} \langle \bar{q}^i(T), \bar{x}^{-1}(T) \rangle \, \bar{q}^i(T) \tag{34}$$

which was to be proven. We will also need to define the forward error residual

$$\varepsilon_N^f(T) = x(T) - [\, x(T-1) \cdots x(T-N)] \bar{f}^N(T) \tag{35}$$

Notice that by (29), (11a), and (12),

$$\begin{aligned} [\bar{F}^N(T)]_T &= [\, x(T-1) \cdots x(T-N)] \bar{f}^N(T) \\ &= x(T) - \varepsilon_N^f(T) \end{aligned} \tag{36}$$

4.B. **THE ONE-STEP BACKWARD PREDICTOR**

It will also be necessary to solve the least squares problem;

$$\left\| \begin{array}{l} N\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right. \\ \\ \\ \\ \end{array} \!\!\! \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline x(0) \\ \vdots \\ x(T-N) \end{bmatrix} - \begin{bmatrix} x(0) & \cdots & 0 \\ x(1) & \cdots & \vdots \\ \vdots & & x(0) \\ \vdots & & \vdots \\ x(T) & & x(T-N+1) \end{bmatrix} \bar{s}^N \right\|_{\Lambda(T+1)}^2 \tag{37}$$

or equivalently

$$\| \bar{x}^N(T+1) - \theta^N(T+1)\bar{s}^N \|^2 \tag{38}$$

Let $\bar{b}^N(T)$ denote the $\bar{s}^N$ that minimizes (38) and let $J_N^b(T)$ be the minimized norm. Thus,

$$J_N^b(T) = \min_{\bar{s}^N} \| \bar{x}^N(T+1) - \theta^N(T+1)\bar{s}^N \|^2 = \| \bar{x}^N(T+1) - \theta^N(T+1)\bar{b}^N(T) \|^2 \tag{39}$$

$\bar{b}^N(T)$ can be interpreted as the one-step backward predictor filter. (38) is minimized just as in the forward predictor problem. Therefore, we have

$$W^N(T+1)\bar{b}^N(T) = \bar{x}_N^{N'}(T+1) \tag{40a}$$

$$J_N^b(T) = ||\bar{x}_N^{N''}(T+1)||^2 \tag{40b}$$

where,

$$Q^\dagger(T+1)\theta^N(T+1) = \begin{array}{r} N \\ \\ T+1-N \end{array} \begin{array}{c} \left\{ \vphantom{\bar{x}} \right. \\ \\ \left\{ \vphantom{\bar{x}} \right. \end{array} \begin{bmatrix} \bar{x}_N^{N'}(T+1) \\ \text{---------} \\ \bar{x}_N^{N''}(T+1) \end{bmatrix} \tag{41}$$

The following vector will be needed

$$\bar{B}^N(T) = \bar{x}^N(T+1) - \theta^N(T+1)\bar{b}^N(T) \tag{42}$$

Just as was done in equations (29) - (34), it is possible to show that

$$\theta^N(T+1)\bar{b}^N(T) = \sum_{i=0}^{N-1} \langle\bar{q}^i(T+1), \bar{x}^N(T+1)\rangle\, \bar{q}^i(T+1) \tag{43}$$

Thus,

$$\bar{B}^N(T) = \bar{x}^N(T+1) - \sum_{i=0}^{N-1} \langle\bar{q}^i(T+1), \bar{x}^N(T+1)\rangle\, \bar{q}^i(T+1) \tag{44}$$

From the Gram-Schmidt procedure, (25), we immediately have

$$\bar{B}^N(T) = \tilde{q}^N(T+1) \tag{45}$$

Thus, from (39)

$$J_N^b(T) = ||\bar{B}^N(T)||^2 = ||\tilde{q}^N(T+1)||^2 \tag{46}$$

Another useful expression for $J_N^b(T)$ is

$$J_N^b(T) = \langle\tilde{q}^N(T+1), \bar{x}^N(T+1)\rangle \tag{47}$$

This follows easily from the Gram-Schmidt procedure, where we have

$$\langle \bar{q}^N(T), \bar{x}^N(T)\rangle = ||\bar{x}^N(T)||^2 - \sum_{i=0}^{N-1} |\langle \bar{q}^i(T), \bar{x}^N(T)\rangle|^2 \tag{48}$$

$$= \sum_{i=N}^{T-1} |\langle q^i(T), x^N(T)\rangle|^2$$

However, from (41)

$$||\bar{x}_N^{N''}(T+1)||^2 = \sum_{i=N}^{T-1} |\langle \bar{q}^i(T), \bar{x}^N(T)\rangle|^2 \tag{49}$$

Thus, (47) follows from (40b), (49), and (48).

We define the backward error residual

$$\varepsilon_N^b(T) = x(T-N) - [\, x(T) \cdots\cdot x(T-N+1)]\bar{b}^N(T) \tag{50}$$

Notice that from (45), (42), (11a), and (12), we have

$$[\tilde{\bar{q}}^N(T+1)]_{T+1} = [\bar{B}^N(T)]_{T+1}$$

$$= x(T-N) - [\, x(T) \cdots\cdot x(T-N+1)]\bar{b}^N(T)$$

$$= \varepsilon_N^b(T) \tag{51}$$

For time updates, we will need the residual

$$\varepsilon_N^{b'}(T) = x(T-N) - [\, x(T) \cdots\cdot x(T-N+1)]\bar{b}^N(T-1) \tag{52}$$

4.C. ORDER UPDATES FOR FORWARD PREDICTOR

From (23a), the N+1-order forward predictor is given by

$$W^{N+1}(T)\bar{f}^{N+1}(T) = \bar{x}^{-1'}_{N+1}(T) \tag{53}$$

From (28), $W^{N+1}(T)$ can be partitioned as

$$W^{N+1}(T) = \begin{array}{c} N\left\{\begin{array}{c} \\ \\ \\ \end{array}\right. \\ \\ T-N-1\left\{\begin{array}{c} \\ \end{array}\right. \end{array} \left[\begin{array}{c|c} & \langle \bar{q}^{o}(T), \bar{x}^{N}(T)\rangle \\ W^{N}(T) & \vdots \\ & \langle \bar{q}^{N-1}(T), \bar{x}^{N}(T)\rangle \\ \hline 0 \cdot \cdot \cdot 0 & \langle \bar{q}^{N}(T), \bar{x}^{N}(T)\rangle \\ \hline \cdot \cdot \cdot 0 & \cdot \cdot \cdot \cdot \end{array}\right] \tag{54}$$

and from the definition of $\bar{x}_N^{N'}(T)$, (41), the above is equivalent to

$$W^{N+1}(T) = \left[\begin{array}{c|c} W^{N}(T) & \bar{x}_N^{N'}(T) \\ \hline 0 \cdot \cdot \cdot 0 & \langle \bar{q}^{N}(T), \bar{x}^{N}(T)\rangle \\ \hline \cdot \cdot \cdot 0 & \cdot \cdot \cdot \cdot \end{array}\right] \tag{55}$$

Also, it is clear from (22) that we can partition $\bar{x}_{N+1}^{-1'}(T)$ as

$$\bar{x}_{N+1}^{-1}(T) = \left[\begin{array}{c} \bar{x}_N^{-1'}(T) \\ \hline \langle \bar{q}^{N}(T), \bar{x}^{-1}(T)\rangle \end{array}\right] \tag{56}$$

To derive order updates, partition $\bar{f}^{N+1}(T)$ as

$$\bar{f}^{N+1}(T) = \left[\begin{array}{c} \bar{f}_N^{N+1}(T) \\ \hline f_{N+1}^{N+1}(T) \end{array}\right] \tag{57}$$

where $\bar{f}_N^{N+1}(T)$ is a vector representing the first N components of $\bar{f}^{N+1}(T)$ and $f_{N+1}^{N+1}(T)$ the last component. Substituting (55), (56), and (57) into (53) leads to one vector equation and one scalar equation to be solved.

$$W^N(T)\bar{f}_N^{N+1}(T) + f_{N+1}^{N+1}(T)\ \bar{x}_N^{N'}(T) = \bar{x}_N^{-1'}(T) \tag{58a}$$

$$f_{N+1}^{N+1}(T) < \bar{q}^N(T),\bar{x}^N(T)> = <\bar{q}^N(T),\bar{x}^{-1}(T)> \tag{58b}$$

Since $W^N(T)$ is invertible, multiplying (58a) on the left by $W^N(T)^{-1}$ and using (23a) and (40a) gives

$$\bar{f}_N^{N+1}(T) = \bar{f}^N(T) - f_{N+1}^{N+1}(T)\bar{b}^N(T-1) \tag{59}$$

$f_{N+1}^{N+1}(T)$ is found from (58b)

$$f_{N+1}^{N+1}(T) = \frac{<\bar{q}^N(T),\bar{x}^{-1}(T)>}{<\bar{q}^N(T),\bar{x}^N(T)>} = \frac{<\tilde{\bar{q}}^N(T),\bar{x}^{-1}(T)>}{<\tilde{\bar{q}}^N(T),\bar{x}^N(T)>}$$

$$= \frac{k^N(T)}{J^b(T-1)} \tag{60}$$

where we have made use of (25c), (47), and the definition

$$k^N(T) = <\tilde{\bar{q}}^N(T),\ \bar{x}^{-1}(T)> \tag{61}$$

Equations (59) and (60) can be combined into

$$\bar{f}^{N+1}(T) = \begin{bmatrix} \bar{f}^N\ (T) \\ \hline 0 \end{bmatrix} + \frac{k^N(T)}{J_N^b(T-1)} \begin{bmatrix} -\bar{b}^N(T-1) \\ \hline 1 \end{bmatrix} \tag{62}$$

Order updates for $J_N^f(T)$ follow easily from (23b) with N replaced by N+1

$$J_{N+1}^f(T) = \left|\left|\bar{x}_{N+1}^{-1''}(T)\right|\right|^2 + \lambda^T|\ x(0)|^2 \tag{63}$$

and from observing that by (22) $\bar{x}_N^{-1''}(T)$ can be partitioned as

$$\bar{x}_N^{-1''}(T) = \left[\begin{array}{c} \langle q^N(T), \bar{x}^{-1}(T)\rangle \\ \hline \bar{x}_{N+1}^{-1''}(T) \end{array}\right] \tag{64}$$

Thus,

$$\begin{aligned} \|\bar{x}_N^{-1''}(T)\|^2 &= |\langle q^N(T), \bar{x}^{-1}(T)\rangle|^2 + \|\bar{x}_{N+1}^{-1''}(T)\|^2 \\ &= \frac{|\langle \tilde{q}^N(T), \bar{x}^{-1}(T)\rangle|^2}{\|\tilde{q}^N(T)\|^2} + \|\bar{x}_{N+1}^{-1''}(T)\|^2 \\ &= \frac{|k^N(T)|^2}{J_N^b(T-1)} + \|\bar{x}_{N+1}^{-1''}(T)\|^2 \end{aligned} \tag{65}$$

where we have used (25c), (46), and (61). Therefore, from (63) and (65) it follows that

$$\begin{aligned} J_{N+1}^f(T) &= \|\bar{x}_N^{-1''}(T)\|^2 + \lambda^T |x(0)|^2 - \frac{|k^N(T)|^2}{J_N^b(T-1)} \\ &= J_N^f(T) - \frac{|k^N(T)|^2}{J_N^b(T-1)} \end{aligned} \tag{66}$$

Order updates for the forward error residual follow by using (62) in the defining equation (35) with N replaced by N + 1, and are given by

$$\varepsilon_{N+1}^f(T) = \varepsilon_N^f(T) - \frac{k^N(T)}{J_N^b(T-1)} \varepsilon_N^b(T-1) \tag{67}$$

4.D. ORDER UPDATES FOR BACKWARD PREDICTOR

In order to derive order updates for the backward predictor, we must start with the least squares problem (38), with N replaced by N + 1. Thus, we have the least squares problem

$$||\bar{x}^{N+1}(T+1) - \theta^{N+1}(T+1)\bar{s}^{N+1}||^2 \tag{68}$$

To derive order updates, partition $\bar{s}^{N+1}$ as

$$\bar{s}^{N+1} = \begin{bmatrix} s_1^{N+1} \\ \hline \bar{s}_N^{N+1} \end{bmatrix} \tag{69}$$

where s_1^{N+1} is the first component of $\bar{s}^{N+1}$ and $\bar{s}_N^{N+1}$ is an N-dimensional column vector consisting of the last N components of $\bar{s}^{N+1}$. From (9a), (10a), and (12), a little thought will show that we can make the following partitioning:

$$\bar{x}^{N+1}(T+1) = \begin{bmatrix} 0 \\ \hline \bar{x}^N(T) \end{bmatrix} \tag{70}$$

$$\theta^{N+1}(T+1) = \left[\begin{array}{c|c} \lambda^{T/2}\, x(0) & 0 \cdot \cdot \cdot \cdot 0 \\ \hline \bar{x}^{-1}(T) & \theta^N(T) \end{array}\right] \tag{71}$$

Substituting (69), (70), and (71) into (68) gives the equivalent least squares problem

$$\left\| \begin{bmatrix} 0 \\ \hline \bar{x}^N\ (T) \end{bmatrix} - \left[\begin{array}{c|c} \lambda^{T/2}\, x(0) & 0 \cdot \cdot \cdot 0 \\ \hline \bar{x}^{-1}(T) & \theta^N(T) \end{array}\right] \begin{bmatrix} s_1^{N+1} \\ \hline \bar{s}_N^{N+1} \end{bmatrix} \right\|^2 \tag{72}$$

The above norm can be minimized with the help of the following T+1-by-T+1 unitary matrix

$$\left[\begin{array}{c|c} 1 & 0 \cdots 0 \\ \hline 0 & \\ \vdots & Q(T) \\ 0 & \end{array}\right] \tag{73}$$

Thus, as was done in subsection 4A, the above unitary transformation of (72) yields the equivalent form

$$\left\| \begin{bmatrix} 0 \\ \hline \bar{x}_N^{N'}(T) \\ \hline \bar{x}_N^{N''}(T) \end{bmatrix} - \begin{bmatrix} s_1^{N+1}\lambda^{T/2}\ \chi(0) \\ \hline s_1^{N+1}\bar{x}_N^{-1'}(T) + W^N(T)\bar{s}_N^{N+1} \\ \hline s_1^{N+1}\bar{x}_N^{-1''}(T) \end{bmatrix} \right\|^2 \tag{74}$$

where we have used (19), (22), and (41). The "middle" part of the above norm can be written out separately so that (74) can be split into

$$\left\| \bar{x}_N^{N'}(T) - s_1^{N+1}\bar{x}_N^{-1'}(T) - W^N(T)\bar{s}_N^{N+1} \right\|^2 + \left\| \begin{bmatrix} 0 \\ \hline \bar{x}_N^{N''}(T) \end{bmatrix} - s_1^{N+1} \begin{bmatrix} \lambda^{T/2}\ \chi(0) \\ \hline \bar{x}_N^{-1''}(T) \end{bmatrix} \right\|^2 \tag{75}$$

Notice that since $\bar{s}_N^{N+1}$ appears only in the first norm of (75), minimization of (75) with respect to $\bar{s}^{N+1}$ implies that the first norm must be minimized with respect to $\bar{s}_N^{N+1}$, regardless of the value of s_1^{N+1} . This is simply an application of the principle of optimality of dynamic programming (ref 10). The principle of optimality can also be used to derive the Levinson

algorithm or similarly the non-adaptive lattice algorithm for the case of known signal statistics (ref 11). Thus, minimization of the first norm gives

$$W^{N}(T)\bar{b}_{N}^{N+1}(T) = \bar{x}_{N}^{N'}(T) - b_{1}^{N+1}\bar{x}_{N}^{-1''}(T) \tag{76}$$

where $\bar{b}^{N+1}(T)$ has been partitioned as

$$\bar{b}^{N+1}(T) = \begin{bmatrix} b_{1}^{N+1}(T) \\ \hline \bar{b}_{N}^{N+1}(T) \end{bmatrix} \tag{77}$$

Multiplying (76) on the left by $W^{N}(T)^{-1}$, and using (23a) and (40a) gives

$$\bar{b}_{N}^{N+1}(T) = \bar{b}^{N}(T-1) - b_{1}^{N+1}\bar{f}^{N}(T) \tag{78}$$

Minimization of the last norm with respect to s_{1}^{N+1} is a simple scalar problem. It is given by the result that for any two vectors x and y

$$\min_{s} ||\bar{x} - s\bar{y}||^{2} = ||\bar{x} - b\bar{y}||^{2} \tag{79a}$$

where

$$b = \frac{\langle\bar{y},\bar{x}\rangle}{\langle\bar{y},\bar{y}\rangle} \tag{79b}$$

Thus,

$$b_{1}^{N+1}(T) = \frac{\langle\bar{x}_{N}^{-1''}(T),\bar{x}_{N}^{N''}(T)\rangle}{||\bar{x}_{N}^{-1''}(T)||^{2} + \lambda^{T}|x(0)|^{2}} \tag{80}$$

From (22),

$$\bar{x}_N^{-1''}(T) = \begin{bmatrix} \langle \bar{q}^N(T), \bar{x}^{-1}(T) \rangle \\ \vdots \\ \langle \bar{q}^{T-1}(T), \bar{x}^{-1}(T) \rangle \end{bmatrix} \tag{81}$$

and from (41) and (27),

$$\bar{x}_N^{N''}(T) = \begin{bmatrix} \langle \bar{q}^N(T), \bar{x}^N(T) \rangle \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{82}$$

Thus,

$$\langle \bar{x}_N^{-1''}(T), \bar{x}_N^{N''}(T) \rangle = \langle \bar{q}^N(T), \bar{x}^{-1}(T) \rangle^* \langle \bar{q}^N(T), \bar{x}^N(T) \rangle$$

$$= \frac{\langle \tilde{\bar{q}}^N(T), \bar{x}^{-1}(T) \rangle^* \langle \tilde{\bar{q}}^N(T), \bar{x}^N(T) \rangle}{\| \tilde{\bar{q}}^N(T) \|^2}$$

$$= k^N(T)^* \tag{83}$$

where we used (46), (47), and (61). Thus, from (83) and (23b), equation (80) is equivalent to

$$b_1^{N+1}(T) = \frac{k^N(T)^*}{J_N^f(T)} \tag{84}$$

Equations (78) and (84) can be combined to yield

$$\bar{b}^{N+1}(T) = \begin{bmatrix} 0 \\ \text{---------} \\ \bar{b}^{N}(T-1) \end{bmatrix} + \frac{k^N(T)^*}{J_N^f(T)} \begin{bmatrix} 1 \\ \text{--------} \\ -\bar{f}^N(T) \end{bmatrix} \tag{85}$$

Order updates for $J_N^b(T)$ are obtained directly from (75) by substituting $\bar{b}^{N+1}(T)$ for $\bar{s}^{N+1}$. Notice that the first norm is zero because of (76). Thus,

$$J_{N+1}^b(T) = \left\| \begin{bmatrix} 0 \\ \text{--------} \\ \bar{x}_N^{N''}(T) \end{bmatrix} - b_1^{N+1}(T) \begin{bmatrix} \lambda^{T/2}\, x(0) \\ \text{-----------} \\ \bar{x}_N^{-1''}(T) \end{bmatrix} \right\|^2$$

$$= \left\|\bar{x}_N^{N''}\ (T)\right\|^2 + \left|b_1^{N+1}(T)\right|^2 \{\lambda^T \left| x(0)\right|^2 + \left\|\bar{x}_N^{-1''}(T)\right\|^2\}$$

$$-b_1^{N+1}(T)\langle\bar{x}_N^{N''}(T),\bar{x}_N^{-1''}(T)\rangle - b_1^{N+1}(T)^*\langle\bar{x}_N^{N''}(T),\bar{x}_N^{-1''}(T)\rangle^* \tag{86}$$

Using (23b), (40b), (83), and (84) in the above, we obtain

$$J_{N+1}^b(T) = J_N^b(T-1) - \frac{|k^N(T)|^2}{J_N^f(T)} \tag{87}$$

Order updates for $\varepsilon_N^b(T)$ are obtained by using (85) and (50)

$$\varepsilon_{N+1}^b(T) = \varepsilon_N^b(T-1) - \frac{k^N(T)^*}{J_N^f(T)} \tag{88}$$

4.E. TIME UPDATES

Order updates for $\varepsilon_N^f(T)$, $\varepsilon_N^b(T)$, $J_N^f(T)$, and $J_N^b(T)$ are given by equations (66), (67), (87), (88) and

$$k^N(T) = \langle \tilde{\bar{q}}(T), \bar{x}^{-1}(T) \rangle \qquad (61)$$

Thus, in order to derive time updates for $\varepsilon_N^f(T)$, $\varepsilon_N^b(T)$, $J_N^f(T)$, and $J_N^b(T)$, it is sufficient to derive the time update equation of $k^N(T)$ for all orders N. To this end, the following T by 1 column vector is defined:

$$\bar{Z}^N(T) = \sum_{i=0}^{N} [\bar{q}^i(T)]_T^* \bar{q}^i(T) \qquad N < T \qquad (89)$$

This section borrows from ref 4, albeit in keeping with the spirit of this technical report, the derivation shows explicitly the role of $\{\bar{q}^0(T) \ . \ . \ . \ \bar{q}^{T-1}(T)\}$. This vector will be useful in deriving time updates for $\tilde{\bar{q}}^N(T)$. It also has the following property:

$$\langle \bar{z}^N(T), \bar{v}(T) \rangle = \begin{cases} [\bar{v}(T)]_T & \bar{v}(T) \varepsilon S^N(T) \\ 0 & \bar{v}(T) \text{ orthogonal to } S^N(T) \end{cases} \qquad (90)$$

This can be seen by observing that from the definition (89) we have

$$\langle \bar{z}^N(T), \bar{q}^i(T) \rangle = \begin{cases} [\bar{q}^i(T)]_T & 0 \leq i \leq N \\ 0 & N < i \end{cases} \qquad (91)$$

and since for any $\bar{v}(T)\varepsilon S^N(T)$, $\bar{v}(T)$ is a linear combination of $\bar{q}^i(T)$, $i = 0,1,\cdots N$, (90) is true.

Order updates for $\bar{z}^N(T)$ follow almost directly from the definition (89)

$$\bar{z}^N(T) = \sum_{i=0}^{N-1} [\bar{q}^i(T)]_T^* \bar{q}^i(T) + [\bar{q}^N(T)]_T^* \bar{q}^N(T)$$

$$= \bar{z}^{N-1}(T) + \frac{\tilde{\bar{q}}^N(T)_T^*}{||\tilde{\bar{q}}^N(T)||^2} \tilde{\bar{q}}^N(T)$$

$$= \bar{z}^{N-1}(T) + \frac{\varepsilon_N^b(T-1)^*}{J_N^b(T-1)} \tilde{\bar{q}}^N(T) \qquad (92)$$

where (46) and (51) have been used.

It will now be shown that

$$\tilde{\bar{q}}^N(T+1) = \left[\frac{\lambda^{1/2}\tilde{\bar{q}}^N(T)}{\varepsilon_N^{b'}(T)} \right] - \varepsilon_N^{b'}(T)\bar{z}^{N-1}(T+1) \qquad (93)$$

The reasoning is as follows: For $0 < i \leq N-1$,

$$\left\langle \left[\frac{\lambda^{1/2}\tilde{\bar{q}}^N(T)}{\varepsilon_N^{b'}(T)} \right], \bar{x}^i(T+1) \right\rangle = \lambda^{1/2} < \tilde{\bar{q}}^N(T), \bar{x}^i(T) > + \varepsilon_N^{b'}(T)^* [\bar{x}^i(T+1)]_{T+1}$$

$$= \varepsilon_N^{b'}(T)^* [\bar{x}^i(T+1)]_{T+1}$$

$$= \varepsilon_N^{b'}(T)^* < \bar{z}^{N-1}(T+1), \bar{x}^i(T+1) > \qquad (94)$$

where we have made use of (90). We thus have from (94) and (26)

$$\left\langle \begin{bmatrix} \lambda^{1/2}\tilde{\bar{q}}^{N}(T) \\ \hline \varepsilon_N^{b'}(T) \end{bmatrix} - \varepsilon_N^{b'}(T)\bar{z}^{N-1}(T+1),\ \bar{q}^{i}(T+1) \right\rangle = 0 \tag{95}$$

for $0 < i \leq N-1$

Also,

$$\begin{bmatrix} \lambda^{1/2}\tilde{\bar{q}}^{N}(T) \\ \hline \varepsilon_N^{b'}(T) \end{bmatrix} = \begin{bmatrix} \lambda^{1/2}\bar{B}^{N}(T-1) \\ \hline \varepsilon_N^{b'}(T) \end{bmatrix}$$

$$= \begin{bmatrix} \lambda^{1/2}\bar{x}^{N}(T) \\ \hline x(T-N) \end{bmatrix} - \begin{bmatrix} \lambda^{1/2}\theta^{N}(T) \\ \hline x(T)\ \cdots\ x(T+1-N) \end{bmatrix} \bar{b}^{N}(T-1)$$

$$= \bar{x}^{N}(T+1) - \theta^{N}(T+1)\bar{b}^{N}(T-1) \tag{96}$$

by (45), (42), (52), and (9b). Thus, (96) and (89) imply

$$\begin{bmatrix} \lambda^{1/2}\tilde{\bar{q}}^{N}(T) \\ \hline \varepsilon_N^{b'}(T) \end{bmatrix} - \varepsilon_N^{b'}(T)\bar{z}^{N-1}(T+1) = \bar{x}^{N}(T+1) - \sum_{i=0}^{N-1} \alpha_i \bar{q}^{i}(T+1) \tag{97}$$

where α_i is found by taking the inner product of the above with $\bar{q}^{i}(T+1)$ and using (95) to yield

$$\langle \bar{x}^{N}(T+1), \bar{q}^{k}(T+1)\rangle - \sum_{i=0}^{N-1} \alpha_i^{*} \langle \bar{q}^{i}(T+1), \bar{q}^{k}(T+1)\rangle = 0$$

$$<\bar{x}^N(T+1),\bar{q}^k(T+1)> - \alpha_k^* <\bar{q}^k(T+1),\bar{q}^k(T+1)> = 0 \tag{98}$$

Thus,

$$\alpha_k = <\bar{q}^k(T+1),\bar{x}^N(T+1)> \tag{99}$$

$$0 < k \leq N-1$$

Substituting (99) into (97) verifies (93).

Time updates for $k^N(T)$ can now be derived. From (61),

$$k^N(T+1) = <\tilde{\bar{q}}^N(T+1),\bar{x}^{-1}(T+1)> \tag{100}$$

$\bar{x}^{-1}(T+1)$ can be partitioned by use of (9b) and (10a)

$$\bar{x}^{-1}(T+1) = \Lambda^{1/2}(T+1)\ \bar{\chi}^{-1}(T+1)$$

$$= \left[\begin{array}{c|c} \lambda^{1/2}\Lambda^{1/2}(T) & 0 \\ \hline 0 & 1 \end{array}\right] \left[\begin{array}{c} \bar{\chi}^{-1}(T) \\ \hline \chi\ (T+1) \end{array}\right]$$

$$= \left[\begin{array}{c} \lambda^{1/2}\bar{x}^{-1}(T) \\ \hline \chi\ (T+1) \end{array}\right] \tag{101}$$

Thus, substitution of (101) and (93) into (100) gives

$$k^N(T+1) = \lambda<\tilde{\bar{q}}^N(T),\bar{x}^{-1}(T)> + \varepsilon_N^{b'}(T)^*\ \chi(T+1)$$

$$- \varepsilon_N^{b'}(T)^*<\bar{z}^{N-1}(T+1),\bar{x}^{-1}(T+1)> \tag{102}$$

From (34), (36), and (90), we have

$$\langle \bar{z}^{N-1}(T+1), \bar{x}^{-1}(T+1)\rangle = \langle \bar{z}^{N-1}(T+1), \bar{F}^{N}(T+1)\rangle$$

$$= [\bar{F}^{N}(T+1)]_{T+1}$$

$$= x(T+1) - \varepsilon_N^f(T+1) \tag{103}$$

Therefore, (102) is simplified to

$$k^N(T+1) = \lambda k^N(T) + \varepsilon_N^{b'}(T)^* \varepsilon_N^f(T+1) \tag{104}$$

Since $\bar{b}^N(T)$ is not actually computed, an equivalent expression for $\varepsilon_N^{b'}(T)$ must be derived that does not depend upon $\bar{b}^N(T)$. Notice that the inner product of $\bar{z}^N(T+1)$ with (93) gives

$$\langle \bar{z}^N(T+1), \tilde{\bar{q}}^N(T+1)\rangle = \varepsilon_N^{b'}(T) - \varepsilon_N^{b'}(T)\langle \bar{z}^N(T+1), \bar{z}^{N-1}(T+1)\rangle \tag{105}$$

From (92), we have

$$\langle \bar{z}^N(T), \bar{z}^{N-1}(T)\rangle = \langle \bar{z}^{N-1}(T), \bar{z}^{N-1}(T)\rangle + \frac{\varepsilon^b(T-1)^*}{J_N^b(T-1)} \langle \tilde{\bar{q}}^N(T), \bar{z}^{N-1}(T)\rangle$$

$$= \langle \bar{z}^{N-1}(T), \bar{z}^{N-1}(T)\rangle$$

$$= \sigma^{N-1}(T-1) \tag{106}$$

where the following definition is made

$$\sigma^N(T-1) = ||\bar{z}^N(T)||^2 \tag{107}$$

Also, from (91) and (51)

$$<\bar{z}^N(T+1),\tilde{\bar{q}}^N(T+1)> = [\tilde{\bar{q}}^N(T+1)]_{T+1}$$

$$= \varepsilon_N^b(T) \tag{108}$$

Therefore, (105) can be written as

$$\varepsilon_N^b(T) = \varepsilon_N^{b'}(T)(1 - \sigma^{N-1}(T)) \tag{109}$$

and substitution into (104) gives the time update

$$k^N(T+1) = \lambda k^N(T) + \frac{\varepsilon_N^b(T)^* \varepsilon_N^f(T+1)}{1 - \sigma^{N-1}(T)} \tag{110}$$

Thus, time updates for $k^N(T)$ is given for all orders N, provided order updates of $\sigma^N(T)$ are derived for all time T. The inner product of $\bar{z}^N(T)$ with (92) yields

$$<\bar{z}^N(T),\bar{z}^N(T)> = <\bar{z}^N(T),\bar{z}^{N-1}(T)> + \frac{\varepsilon_N^b(T-1)^*[\tilde{\bar{q}}^N(T)]_T}{J_N^b(T-1)}$$

$$= <\bar{z}^{N-1}(T),\bar{z}^{N-1}(T)> + \frac{|\varepsilon_N^b(T-1)|^2}{J_N^b(T-1)} \tag{111}$$

where we have used (106) and (51). Therefore

$$\sigma^{N}(T-1) = \sigma^{N-1}(T-1) + \frac{|\varepsilon_N^b(T-1)|^2}{J_N^b(T-1)} \tag{112}$$

5. THE LEAST SQUARES LATTICE ALGORITHM

Equations (66), (67), (87), (88), (110), and (112) constitute the lattice algorithms. The variables $\varepsilon_N^f(T)$, $\varepsilon_N^b(T)$, $J_N^f(T)$, $J_N^b(T)$ are easily initialized at N=0 by considering the least squares problems (17) and (38) with N=0, as well as (36) and (51). Thus,

$$J_o^f(T) = ||\bar{x}^o(T+1)||^2$$

$$\varepsilon_o^f(T) = x(T)$$

$$J_o^b(T) = ||\bar{x}^o(T+1)||^2$$

$$\varepsilon_o^b(T) = x(T) \tag{113}$$

However, as was done in (101), we have

$$\bar{x}^o(T+1) = \begin{bmatrix} \lambda^{\frac{1}{2}}\bar{x}^o(T) \\ \text{-------} \\ x(T) \end{bmatrix} \tag{114}$$

and therefore

$$||\bar{x}^o(T+1)||^2 = \lambda||\bar{x}^o(T)||^2 + |x(T)|^2 \tag{115}$$

The initializations of (113) can be written as

$$\varepsilon_o^f(T) = \varepsilon_o^b(T) = x(T)$$

$$J_o^f(T) = J_o^b(T) = \lambda J_o^b(T-1) + |x(T)|^2 \tag{116}$$

From (89) we have for N=0

$$||\bar{z}^o(T)||^2 = |[\bar{q}^o(T)]_T|^2$$

$$= \frac{|[\tilde{\bar{q}}^{o}(T)]_T|^2}{||\tilde{\bar{q}}^{o}(T)||^2}$$

$$= \frac{|\varepsilon_o^b(T-1)|^2}{J_o^b(T-1)} \tag{117}$$

Thus,

$$\sigma^{o}(T-1) = \frac{|\varepsilon_o^b(T-1)|^2}{J_o^b(T-1)} \tag{118}$$

However, notice that (118) can be given by (112) if we define

$$\sigma^{-1}(T-1) = 0 \tag{119}$$

The least squares lattice algorithm is summarized below.

$$\varepsilon_o^f(T) = \varepsilon_o^b(T) = x(T) \tag{120a}$$

$$J_o^f(T) = J_o^b(T) = \lambda J_o^b(T-1) + |x(T)|^2 \tag{120b}$$

$$\sigma^{-1}(T) = 0 \tag{120c}$$

For $N=0,1,\cdots N_o \leq T$

$$k^N(T) = \lambda k^N(T-1) + \frac{\varepsilon_N^b(T-1)^* \varepsilon_N^f(T)}{1-\sigma^{N-1}(T-1)} \tag{121a}$$

$$\sigma^N(T-1) = \sigma^{N-1}(T-1) + \frac{|\varepsilon_N^b(T-1)|^2}{J_N^b(T-1)} \tag{121b}$$

$$\varepsilon_{N+1}^f(T) = \varepsilon_N^f(T) - \frac{k^N(T)\varepsilon_N^b(T-1)}{J_N^b(T-1)} \tag{121c}$$

$$\varepsilon^b_{N+1}(T) = \varepsilon^b_N(T-1) - \frac{k^N(T)^* \varepsilon^f_N(T)}{J^f_N(T)} \tag{121d}$$

$$J^f_{N+1}(T) = J^f_N(T) - \frac{\left|k^N(T)\right|^2}{J^b_N(T-1)} \tag{121e}$$

$$J^b_{N+1}(T) = J^b_N(T-1) - \frac{\left|k^N(T)\right|^2}{J^f_N(T)} \tag{121f}$$

In order to prevent division by zero, it is recommended that $J^b_o(-1)$, $J^f_o(-1)$ be initialized to some small positive number δ.

$$J^b_o(-1) = J^f_o(-1) = \delta \tag{122}$$

Also, the algorithm assumes that $k^N(T)$ is initialized to zero until it is used in the update (121a).

6. SOLUTION OF THE GENERAL LEAST SQUARES PROBLEM

We are now in a position to obtain order updates to the minimization of

$$||\bar{y}(T+1) - \theta^N(T+1)\bar{s}^N||^2 \tag{123}$$

First, we need the general solution to the N^{th}-order problem, along with some useful properties. Just as for the forward and backward predictor, it can be shown that (123) is equivalent to

$$||\bar{y}_N'(T+1) - W^N(T+1)\bar{s}^N||^2 + ||\bar{y}_N''(T+1)||^2 \tag{124}$$

where

$$Q^\dagger(T+1)\bar{y}(T+1) = \begin{array}{r} N \left\{ \right. \\ \\ T+1-N \left\{ \right. \end{array} \begin{bmatrix} y_N'(T+1) \\ ------- \\ \bar{y}_N''(T+1) \end{bmatrix} \tag{125}$$

and minimization of (124) is given by $\bar{s}^N = \bar{h}^N(T)$, where

$$W^N(T+1)\bar{h}^N(T) = \bar{y}_N'(T+1) \tag{126a}$$

and the minimum value is

$$J_N(T) = ||\bar{y}_N''(T+1)||^2 \tag{126b}$$

Also, we define the following T+1 by 1 column vector

$$\bar{H}^N(T) = \theta^N(T+1)\bar{h}^N(T) \tag{127}$$

Just as in equations (29) - (34), we have

$$\bar{H}^N(T) = \sum_{i=0}^{N-1} \langle\bar{q}^i(T+1),\bar{y}(T+1)\rangle\bar{q}^i(T+1) \tag{128a}$$

and consequently

$$\bar{y}(T+1) - \bar{H}^N(T) = \bar{y}(T+1) - \sum_{i=N}^{T} \langle \bar{q}^i(T+1), \bar{y}(T+1)\rangle \bar{q}^i(T+1) \tag{128b}$$

Order updates are derived in exactly the same fashion as for the forward predictor, but with $\bar{q}^i(T)$, $\bar{x}^{-1}(T)$, $\bar{f}^N(T)$, and $\bar{W}^N(T)$ replaced by $\bar{q}^i(T+1)$, $\bar{y}(T+1)$, $\bar{h}^N(T)$, and $\bar{W}^N(T+1)$, respectively. Therefore,

$$\bar{h}^{N+1}(T) = \begin{bmatrix} \bar{h}^N(T) \\ \hdashline 0 \end{bmatrix} + \frac{\hat{k}^N(T)}{J_N^b(T)} \begin{bmatrix} -\bar{b}^N(T) \\ \hdashline 1 \end{bmatrix} \tag{129a}$$

$$J_{N+1}(T) = J_N(T) - \frac{|\hat{k}^N(T)|^2}{J_N^b(T)} \tag{129b}$$

where,

$$\hat{k}^N(T) = \langle \tilde{\bar{q}}^N(T+1), y(T+1)\rangle \tag{130}$$

Order updates for $\varepsilon_N(T)$ and $\varepsilon_N'(T)$ are easily obtained by use of (129a) with (15a), (15b), (50), (52), and (109)

$$\varepsilon_{N+1}(T) = \varepsilon_N(T) - \frac{\hat{k}^N(T)}{J_N^b(T)} \varepsilon_N^b(T) \tag{131a}$$

$$\varepsilon_{N+1}'(T) = \varepsilon_N'(T) - \frac{\hat{k}^N(T-1)}{J_N^b(T-1)} \cdot \frac{\varepsilon_N^b(T)}{1-\sigma^{N-1}(T)} \tag{131b}$$

Time updates for $\hat{k}^N(T)$ are derived in the same manner as for $k^N(T)$. By using (93) and proceeding as in equation (95), with $\bar{x}^{-1}(T+1)$ replaced by $\bar{y}(T+1)$, we obtain

$$\hat{k}^N(T) = \lambda \hat{k}^N(T-1) + \varepsilon_N^{b'}(T)^* y(T+1)$$

$$- \varepsilon_N^{b'}(T)^* < \bar{z}^{N-1}(T+1),\ \bar{y}(T+1)> \qquad (132)$$

From (128a), we have

$$<\bar{z}^{N-1}(T+1),\ \bar{y}(T+1)> = [H^N(T)]_{T+1}$$

$$= [\ x(T) \cdots x(T-N+1)]\bar{h}^N(T)$$

$$= \ y(T) - \varepsilon_N(T) \qquad (133)$$

Therefore, using (133) and (109) with (132) gives

$$\hat{k}^N(T) = \lambda \hat{k}^N(T-1) + \frac{\varepsilon_N^b(T)^* \varepsilon_N(T)}{1-\sigma^{N-1}(T)} \qquad (134)$$

Also, from (128b) it is possible to show in a way similar to that in equations (93) - (99) that

$$\bar{y}(T+1) - \bar{H}^N(T) = \left[\frac{\lambda^{\frac{1}{2}}(\bar{y}(T)-\bar{H}^N(T-1))}{\varepsilon_N'(T)} \right] - \varepsilon_N'(T)\bar{z}^{N-1}(T+1) \qquad (135)$$

The inner product of $\overline{z}^N(T+1)$ with (135) yields

$$\varepsilon_N(T) = \varepsilon_N'(T)\ (1-\sigma^{N-1}(T)) \tag{136}$$

Thus, substitution of (136) into (134) gives an equivalent time update for $\hat{k}^N(T)$ as a function of $\varepsilon_N'(T)$.

$$\hat{k}^N(T) = \lambda\hat{k}^N(T-1) + \varepsilon_N^b(T)^*\varepsilon_N'(T) \tag{137}$$

Thus, order updates are given by (131a, b) with time updates (134) and (137). From (123) it is quite clear that $\varepsilon_N(T)$ and $\varepsilon_N'(T)$ are initialized as

$$\varepsilon_o(T) = \varepsilon_o'(T) = \ y(T) \tag{138}$$

7. NUMERICAL CONDITIONING

It is well known that in solving ill-conditioned least squares problems, some algorithms may prove to be numerically superior to others in the sense that they may need only half the word length to give the same result, or equivalently, can provide far greater numerical precision with the same word length. A classic example is that of the Kalman filter and the so-called square-root filter (ref 12, 13). Both algorithms solve the same least squares problem, but the square-root filter is numerically superior for ill-conditioned problems. This is because the Kalman filter can be shown to be based upon the pseudoinverse method of solving the normal equation associated with least squares, whereas the square-root filter is based upon using a unitary transformation to solve the least squares problem, as outlined in Section 4.A. The use of this unitary transformation (or also called a Householder transformation) leads to a matrix equation involving the Cholesky decomposition of a covariance matrix, whereas the normal equation involves the covariance matrix directly. The condition number of the covariance matrix is the square of the condition number of the Cholesky decomposition, and thus the square-root filter is better conditioned. The Cholesky decomposition of a matrix R is S, where $S^{\dagger}S = R$. S looks like a generalized "square root" and hence the name of the filter. See ref 14 for further details.

It at first might appear that the least squares lattice filter is similar to the square-root filter. For instance, equations (23a), (40a), and (126a) are of the general form

$$W^N(T)\bar{U}^N = \bar{V}^N \tag{139}$$

$W^N(T)$ is actually a Cholesky decomposition (ref 14) of the sample auto-covariance matrix

$$R^N(T) = \Theta^N(T)^\dagger \Theta^N(T) \tag{140}$$

This is seen as follows

$$\begin{aligned} R^N(T) &= \Theta^N(T)^\dagger \Theta^N(T) \\ &= \Theta^N(T)^\dagger Q(T) Q^\dagger(T) \Theta^N(T) \\ &= [Q^\dagger(T)\Theta^N(T)]^\dagger Q^\dagger(T)\Theta^N(T) \\ &= W^N(T)^\dagger W^N(T) \end{aligned} \tag{141}$$

The condition number (ref 14) of (139) is thus given by

$$\sqrt{\frac{\eta_{max}}{\eta_{min}}} \tag{142}$$

where η_{max} and η_{min} are, respectively, the maximum and minimum eigenvalues of R. This is to be contrasted with the pseudoinverse method of solving least squares problems, which leads to normal equations of the type

$$R^N(T)\bar{U}^N = \bar{r} \tag{143a}$$

with condition number

$$\frac{\eta_{max}}{\eta_{min}} \tag{143b}$$

The well-conditioned property of (139) is not maintained because of the way in which $f_{N+1}^{N+1}(T)$ and $b_1^{N+1}(T)$ are updated. For example, from equation (60) we have

$$f_{N+1}^{N+1}(T) = \frac{\langle \bar{q}^N(T), \bar{x}^{-1}(T) \rangle}{\langle \bar{q}^N(T), x^N(T) \rangle} \tag{144}$$

However, time updates were not derived for $\bar{q}^N(T)$, but rather the numerator and denominator of (144) are multiplied by $||\tilde{\bar{q}}^N(T)||$ to yield

$$f_{N+1}^{N+1}(T) = \frac{\langle \tilde{\bar{q}}^N(T), \bar{x}^{-1}(T) \rangle}{\langle \tilde{\bar{q}}^N(T), \bar{x}^N(T) \rangle} \tag{145}$$

and updates are then derived for $\tilde{\bar{q}}^N(T)$ and consequently $k^N(T)$ and $J_N^b(T-1)$. It is for this reason that the least squares lattice filter does not have condition number $\sqrt{\eta_{max}/\eta_{min}}$. Since the algorithm is an order and time recursive solution to a normal equation (ref 2), it can be concluded that the condition number is η_{max}/η_{min}.

The normalized least squares lattice filter will not be derived here. It further requires time updates for $\sigma^N(T)$, $J_N(T)$, $J_N^b(T)$ and $J_N^f(T)$. Suffice it to say that $\varepsilon_N(T)$, $\varepsilon_N^f(T)$ and $\varepsilon_N^b(T)$ are normalized such that their magnitudes are less than one. Also the geometric mean of the reflection coefficients, $k^N(T)/J_N^b(T-1)$ and $k^N(T)/J_N^f(T)$, is used and is in magnitude less than one. The net result is an algorithm with fewer update equations and with variables in magnitude less than one except for an "unnormalization" to obtain the desired residuals, which involves updates for quantities of magnitude greater than one. This normalized algorithm may be better suited for fixed-point arithmetic, although there is still at least one update equation in which overflow can occur (ref 3).

The normalized algorithm is erroneously considered to have better numerical conditioning than the unnormalized algorithm. Despite the normalization of the variables, the normalized algorithm is still based upon updates for $\tilde{\bar{q}}^N(T)$ and not $\bar{q}^N(T)$, and thus has the same numerical conditioning as the unnormalized algorithm.

A simple numerical example can illustrate the conditioning. Consider a second-order problem to solve the weighted least squares problem of (13) with $\chi(T)$ and $y(T)$ equal to one for all T. We thus have the least squares problem

$$\left\| \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & 1 \end{bmatrix} \bar{s}^2 \right\|^2_{\Lambda(T+1)} \tag{146}$$

or

$$\left\| \begin{bmatrix} \lambda^{(T-1)/2} \\ \lambda^{(T-2)/2} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \lambda^{\frac{1}{2}} \\ 1 \end{bmatrix} - \begin{bmatrix} \lambda^{(T-1)/2} & 0 \\ \lambda^{(T-2)/2} & \lambda^{(T-2)/2} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \lambda^{\frac{1}{2}} & \lambda^{\frac{1}{2}} \\ 1 & 1 \end{bmatrix} \bar{s}^2 \right\|^2 \tag{147}$$

Notice that for $\lambda < 1$, $\lambda^{(T-1)/2} \to 0$ for $T \to \infty$, and the two columns of the above matrix tend to be "less" numerically independent as T increases. More specifically, $\Theta^N(T)$, N = 2, becomes more ill-conditioned as T increases. As a result, various quantities such as $J_1^b(T) \to 0$ for $T \to \infty$ and division by zero will occur because of finite word length. All that is needed to investigate

the conditioning of the various algorithms is to solve the least squares problem for $\lambda < 1$ on a digital computer and observe the time T at which a division by zero is attempted. One will find that a division by zero is attempted for the unnormalized and normalized least squares lattice at the same time T, whereas the "square-root" type algorithm based upon a unitary transformation outlined earlier will operate for twice as long before a division by zero occurs.

8. SUMMARY

A new derivation of the least squares lattice filter is given in this report. This derivation shows that the least squares lattices (both unnormalized and normalized) have the same numerical conditioning as in the case of solving the least squares problem by the normal equation. Thus it is incorrect to consider that, for ill-conditioned problems, the normalized lattice is superior to the unnormalized lattice in the same sense that the square-root filter is numerically superior to the Kalman filter.

9. REFERENCES

1. M. Morf, D. Lee, "Recursive Least Squares Ladder Forms for Fast Parameter Tracking," Proc. of the 1978 IEEE Conf. on Decision and Control, Jan. 10-12, 1979, San Diego, CA. pp 1362-1367.

2. J.D. Pack, E.H. Satorius, "Least Squares, Adaptive Lattice Algorithms," Naval Ocean Systems Center Technical Report, TR 423, April 1979.

3. D. Lee, M. Morf, and B. Friedlander, "Recursive Least Squares Ladder Estimation Algorithms," IEEE Trans. Acoustics, Speech, and Signal Processing, vol. ASSP-29, no. 3, pp 627-641, June 1981.

4. M.J. Shensa, "Recursive Least Squares Lattice Algorithms: A Geometrical Approach," IEEE Trans. Automatic Control, vol. AC-26, pp 695-702, June 1981.

5. P. Businger, G.H. Golub, "Linear Least Squares Solution by Householder Transformation," Numerische Mathematik, vol. 7, no. 7, 1965.

6. A. Bjorck, "Solving Linear Least Squares Problems by Gram-Schmidt Orthogonalization," BIT, vol. 7, pp 1-21, 1967.

7. B. Friedlander, "Lattice Filters for Adaptive Processing," Proceedings of the IEEE vol. 70, no. 8, August 1982.

8. E.H. Satorius, J.D. Pack, "Applications of Least Squares Lattice Algorithms to Adaptive Equalization," IEEE Trans. on Comm., vol. COM-29, pp 136-142, Feb. 1981.

9. E.H. Satorius, M.J. Shensa, "On the Application of Recursive Least Squares Methods to Adaptive Processing," International Workshop on Applications of Adaptive Control, Yale University, New Haven, Conn., August 1979.

10. R. Bellman, S. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1962.

11. F. Pollara, K. Yao, "Dynamic Programming, Levinson Algorithm, and Lattice Equalizers," Proc. of the 20th Allerton conf. on Communication, Control and Computing. Oct. 1982. pp. 507-515.

12. P. Dyer and S.M. Reynolds, "Extension of Square Root Filtering to Include Process Noise," Journal of Optimization Theory and Applications, vol. 3, no. 6, 1969.

13. P. Kaminski, A. Bryson, S. Schmidt, "Discrete Square Root Filtering: A Survey of Current Techniques," IEEE Trans. on Automatic Control, vol. AC-16, no. 6, Dec. 1971.

14. C. Lawson, R. Hanson, Solving Least Squares Problems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.